Bài Tập 4: Xuất Dữ Liệu

Mục Lục

[Phần 1: Tải ảnh bạn muốn xuất thông tin 2](#_Toc459912441)

[Phần 2: (không bắt buộc) Tạo biểu đồ/histogram 2](#_Toc459912442)

[Phần 3: Xuất Dữ Liệu 2](#_Toc459912443)

[Phần 4: Truy Xuất Video 6](#_Toc459912444)

[*Phần 5: (không bắt buộc)* Các Đối Tượng và Hàm khác 7](#_Toc459912445)

Phần 1: Tải ảnh bạn muốn xuất thông tin

Trong bài tập về xuất dữ liệu này, chúng ta sẽ sử dụng tập dữ liệu: Độ Cao Tán Cây (Canopy Height) 2005 trích xuất từ dữ liệu lidar đo trên máy bay/airborn lidar - Geoscience Laser Altimeter System (GLAS) và dữ liệu không gian địa lý.

* + 1. Copy và paste những dòng lệnh sau vào cửa sổ code editor. Click Run để tìm hiểu tập dữ liệu.

**var** CanopyHeight **=** ee**.**Image**(**"NASA/JPL/global\_forest\_canopy\_height\_2005"**);**

Map**.**addLayer**(**CanopyHeight**,**

**{**min**:** 0**,** max**:** 36**,** palette**:** **[**'FFFFFF'**,** '00FF00'**]},**

'canopy height'**);**

Map**.**setCenter**(**100.096435546875**,** 13.966054081318301**,** 8**);**

* + 1. Tiếp theo, sử dụng công cụ draw để vẽ một vùng/polygon nhỏ đại diện cho vùng mà bạn muốn trích xuất thông tin về độ cao tán cây. Tham khảo bài tập 3, Phần 2 nếu bạn cần ôn lại cách số hóa một đối tượng hình học/geometry. Hãy nhớ rằng: vùng càng nhỏ thì thời gian tải càng nhanh.

Phần 2: (không bắt buộc) Tạo biểu đồ/histogram

* + 1. Script/tập lệnh dưới đây giúp tạo một biểu đồ độ cao tán cây cho toàn bộ khu vực nghiên cứu mà bạn vừa số hóa. Copy và paste tập lệnh này vào cửa sổ code editor để biết được/investigate khoảng độ cao của tán cây trong khu vực nghiên cứu của mình.

// Generate the histogram data.

var CanopyHeightHistogram = Chart.image.histogram(CanopyHeight, geometry);

CanopyHeightHistogram = CanopyHeightHistogram.setOptions({

title: 'Histogram of Canopy Height'

});

print(CanopyHeightHistogram);

Phần 3: Xuất Dữ Liệu

Xuất dữ liệu được thực hiện với Code Editor thông qua các hàm export/xuất, những hàm này gồm nhiều lựa chọn để xuất các dữ liệu dạng ảnh/image, bảng/table, và video. Bạn sẽ tập trung vào hàm **Export.image.toDrive()** để tải tập dữ liệu ảnh. Bạn cũng có thể xuất tập ảnh của mình về máy tính cá nhân hoặc lưu tại Google Cloud.

Các hàm export gồm các đối số/argument tùy chọn giúp bạn có thể kiểm soát một số đặc tính quan trọng trong kết quả xuất dữ liệu cuối cùng của mình, ví dụ như độ phân giải/resolution và lưới chiếu/projection.

* 1. Xem Chỉ Dẫn Về Hàm Export.image.toDrive()
     1. Dưới tab **Docs**, **tìm đến** và **mở** chỉ dẫn củahàm **Export.image.toDrive()**, nằm trong nhóm Export. **Đọc kỹ** những chỉ dẫn này.
  2. Tạo một tác vụ xuất dữ liệu/export task
     1. **Thêm** những dòng lệnh sau đây xuống dưới cùng phần script/tập lệnh bạn đang làm việc. Những dòng lệnh này sẽ tạo ra một tác vụ/task trong tab task mà bạn có thể sử dụng để xuất ảnh mình muốn. Xuất các ảnh và lưu tại Google Drive của bạn. **Tham khảo** hộp ghi chú phía dưới đây để hiểu thêm về các tham số cần được xác định.

//Export the image

Export**.image(**CanopyHeight**,** "MyFirstExport"**,**

**{**maxPixels**:** 1e13**,**

region**:** geometry**,**

crs**:** 'EPSG:32647'**,**

scale**:** 1000**});**

**Ghi chú –** Trong ví dụ này, bạn đã xác định một vài đối số tùy chọn được nhận diện bởi **Export.image()**. Mặc dù hàm này bao gồm các tham số tùy chọn (không bắt buộc), nhưng làm quen với những tham số này rất hữu ích cho bạn:

**maxPixels** – Tham số này giới hạn số lượng pixels trong ảnh được truy xuất. Theo mặc định/default, giá trị này là 100,000,000 pixel. Bạn có thể gán cho thông số này một giá trị lớn hơn hoặc nhỏ hơn giới hạn mặc định/defaul. “1e13” hay 10 mũ 13 (1013) và là số lượng pixel lớn nhât mà bạn có thể xuất trong GEE.

**region –** Theo mặc định, dữ liệu được xuất có phạm vi bằng độ rộng cửa sổ view của Code Editor tuy nhiên bạn cũng có thể dùng một đối tượng hình học để thay đổi phạm vi xuất kết quả.

**crs –** Là hệ tọa độ của ảnh kết quả. Tham số này được xác định bằng code EPSG. Bạn có thể tra cứu các code **EPSG** code cho hệ tọa độ mà bạn muốn sử dụng tại <http://spatialreference.org>.

**scale –** Là độ phân giải tính theo mét (m) của mỗi pixel. Độ phân giải của dữ liệu độ cao tán cây là 30 arc-seconds/giây góc hoặc tương đương với 1 kilomet (km)

* 1. Xuất ảnh vào Google Drive với export task

**Ghi chú về Google Drive:** Google Drive là nơi lưu trữ tạm thời cho bất kỳ dữ liệu nào mà bạn tải về. Để chia sẻ với những người khác, bạn nên sử dụng các giải pháp liên kết/corporate solutions, như email hoặc CloudVault

* + 1. **Chạy/Run** tập lệnh mà chúng ta vừa copy vào GEE. Sau đó, tab **Tasks** phía trên bên phải của Code Editor sẽ được được tô đậm.

Highlighted tasks tab

* + 1. **Click** vào tab **Tasks**. Sau đó **click** vào nút **Run** màu xanh biển (như hình dưới) để dữ liệu được xuất tại Google Drive của bạn.

Run your first export.

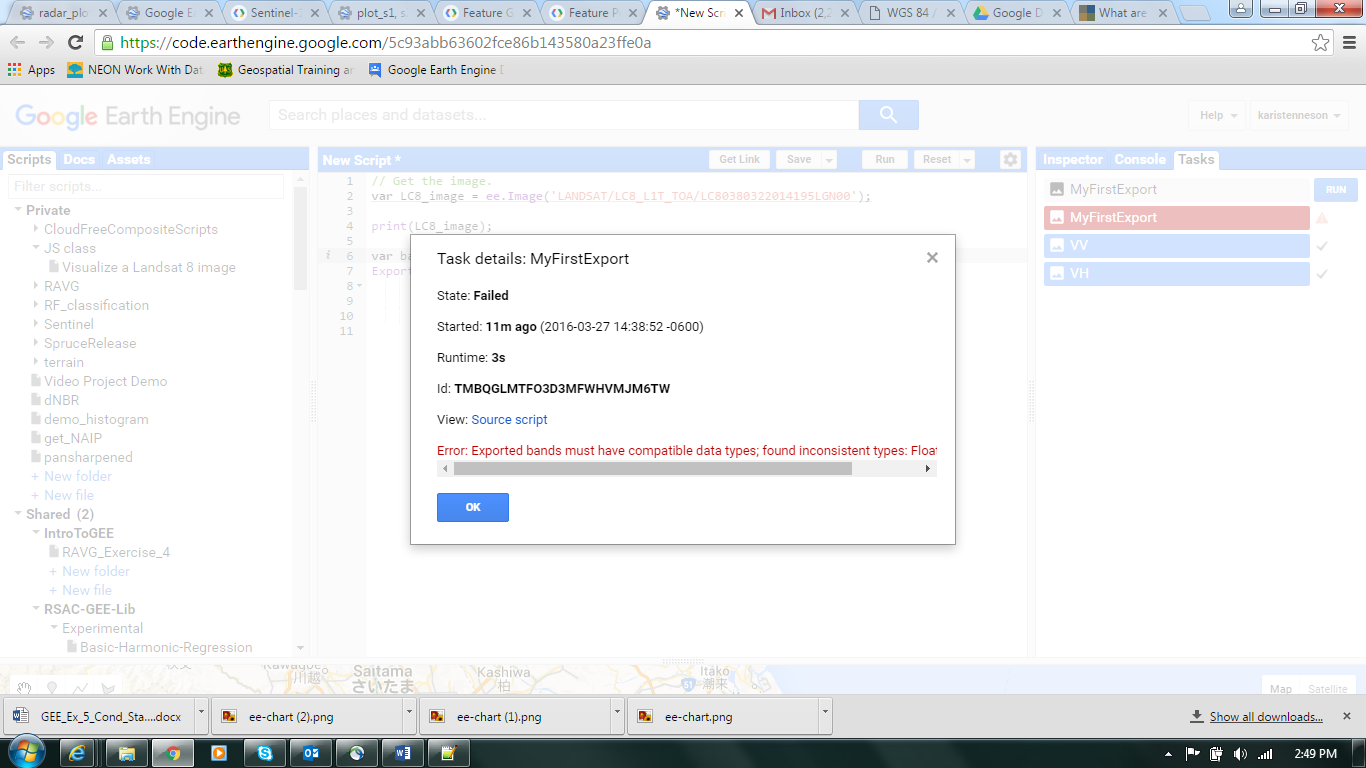
* + 1. **Đọc kỹ** thông tin tại cửa sổ **Initiate export** vừa xuất hiện (dưới đây). Sau đó **click Run** để bắt đầu.

**Ghi chú:** Chức năng/Task này sẽ được xuất kết quả đến Google Drive trong Google Account của bạn. Đó là ảnh GeoTiff có độ phân giải 1000 mét với tên là **MyFirstExport**.

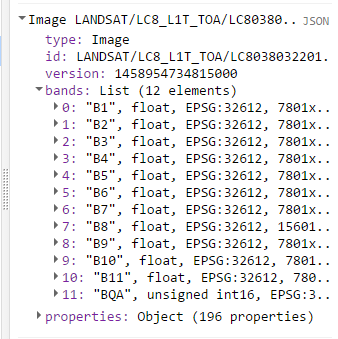
* + 1. Sau khi click **Run** ở phía dưới cùng cửa sổ Initiate export để bắt đầu xuất dữ liệu, màn hình của bạn sẽ báo hiệu quá trình xuất dữ liệu đang được thực hiện. Task/tác vụ phía dưới tab Task của Code Editor có biểu tượng GEE quay tròn ngay bên cạnh. Sẽ mất một khoảng thời gian nhất định để task này được thực hiện. Khi quy trình hoàn thành, biểu tượng quay tròn sẽ biến mất và tên của task sẽ chuyển thành màu xanh lục. **Kiểm tra** xem task trên màn hình của bạn đã chuyển sang xanh.



**Ghi chú:** Nếu bạn thử xuất cảnh ảnh Landsat mà không giới hạn/sub-set ảnh trước để có các band/phổ ảnh mà mình muốn, bạn sẽ nhận được thông báo lỗi.



Đó là bởi vì các band/kênh phổ của ảnh Lansdat không đều được lưu dưới cùng một dạng dữ liệu. Phần lớn các band/kênh phổ ở dạng 32 bit float (float 32), nhưng BQA lại ở dạng unsigned 16 bit integer (UInt16). Bạn có thể tra cứu dạng dữ liệu của mỗi band/kênh phổ bằng hàm print (như hình dưới đây). Bạn có thể truy xuất 10 band/kênh phổ ảnh đầu tiên trước hoặc chuyển band không cùng định dạng, BQA, về cùng dạng dữ liệu với các band khác (UInt16 thành float32).



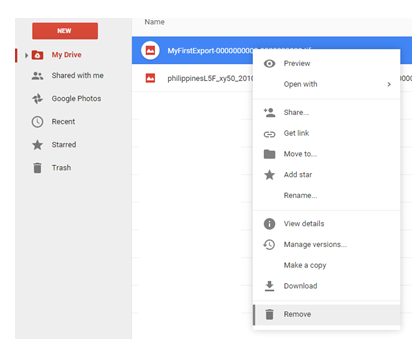
* 1. Đọc kết quả
     1. Khi quá trình xuất/export hoàn thành, mở Google Drive của bạn sử dụng link dưới đây:

<https://drive.google.com/drive/my-drive>

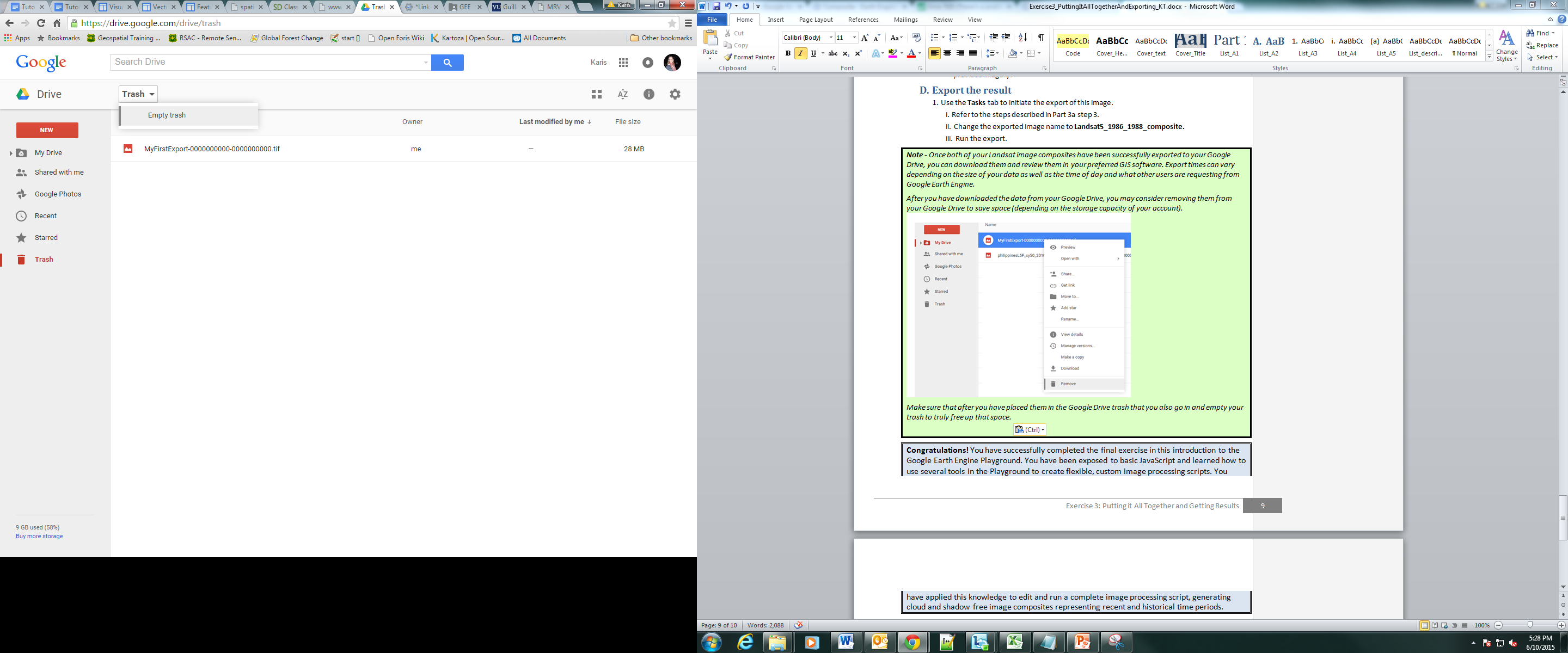
* + 1. **Locate/Lưu** những ảnh mới, có thể gọi tên tương tự như **MyFirstExport-0000000000-0000000000.tif**, tại Google Drive của bạn.
    2. *Không bắt buộc –* **Tải** một trong những ảnh và **xem** ảnh bằng phần mềm GIS mà bạn muốn (ví dụ, ArcMap hoặc QGIS).

***Ghi chú –*** *Một khi cả hai tổ hợp ảnh Landsat của bạn được lưu xuất thành công vào Google Drive, bạn có thể tải và xem lại chúng bằng các phần mềm GIS mà bạn muốn. Thời gian lưu xuất có thể thay đổi tùy thuộc vào độ lớn dữ liệu của bạn cũng như thời gian trong ngày và những gì mà những người dùng khác đang thực hiện trên Google Earth Engine.*

***Ghi Chú Về Quản Lý Dữ Liệu Trong Google Drive –*** *Sau khi bạn đã tải dữ liệu từ Google Drive, bạn có thể nghĩ tới việc xóa chúng khỏi Google Drive để tiết kiệm bộ nhớ (phụ thuộc vào sức chứa của tài khoản của bạn).*



*Sau khi bạn chuyển dữ liệu đến thùng rác của Google Drive, bạn hãy vào thùng rác và làm trống để bộ nhớ được xóa hoàn toàn.*



* 1. Lưu sript/tập lệnh đã hoàn thành vào tài khoản GEE của bạn
     1. **Click** vào nút **Save** trong cửa sổ **Code editor**.
     2. **Nhập** một tên mới hoặc **giữ nguyên** tên mặc định và **click** nút **okay**.

**Ghi chú –** Ví dụ trên tập trung vào một task/tác vụ mà sẽ được lưu xuất vào Google Drive trong tài khoản Google của bạn. Mỗi lần tải sẽ có một ảnh GeoTiff có độ phân giải 30 mét. Với Palawan, mỗi vùng tổ hợp được truy xuất có độ lớn khoảng 370 MB. Đảm bảo rằng bạn có khoảng trống trong Google Drive! Nếu cần có thể xóa và làm trống thùng rác để có thêm bộ nới

Phần 4: Xuất Video

* 1. Tập chỉnh sửa một script/tập lệnh
     1. Đọc kỹ dòng code dưới đây và chạy để xem dòng code này thực hiện gì. Sử dụng những gì bạn đã học cho tới bây giờ.
     2. Chỉnh sửa nó để lưu xuất một video của những ảnh Landsat cho khu vực nghiên cứu của bạn tại vùng Mekong

// Video demo

**var** geometry **=** /\* color: d63000 \*/ee**.**Geometry**.**Polygon**(**

**[[[-**112.07496643066406**,** 40.91892834534881**],**

**[-**111.83326721191406**,** 40.918409488474005**],**

**[-**111.83326721191406**,** 41.06767190734072**],**

**[-**112.07290649414062**,** 41.06818959082366**]]]);**

// load Landsat 5 imagery

**var** collection **=** ee**.**ImageCollection**(**'LT5\_L1T\_TOA'**)**

// Wasatch Front

**.**filterMetadata**(**'WRS\_PATH'**,** 'equals'**,** 38**)**

**.**filterMetadata**(**'WRS\_ROW'**,** 'equals'**,** 32**)**

// filter cloudy scenes

**.**filterMetadata**(**'CLOUD\_COVER'**,** 'less\_than'**,** 10**)**

// 20 years

**.**filterDate**(**'1984-01-01'**,**'2012-12-30'**)**

// need to have 3-band imagery

**.select([**'B7'**,** 'B4'**,** 'B3'**])**

// need to make it 8-bit

**.**map**(function(image)** **{**

**return** **image.**multiply**(**512**).**uint8**();**

**});**

Map**.**centerObject**(**geometry**,** 10**);**

Map**.**addLayer**(**ee**.**Image**(**collection**.**first**()));**

// Export (change dimensions or scale for higher quality)

Export**.**video**(**collection**,** 'example\_video'**,** **{**

dimensions**:** '720'**,**

framesPerSecond**:** 12**,**

region**:** JSON**.**stringify**(**geometry**.**getInfo**())});**

Phần 5*: (không bắt buộc)* Các Đối Tượng và Hàm khác

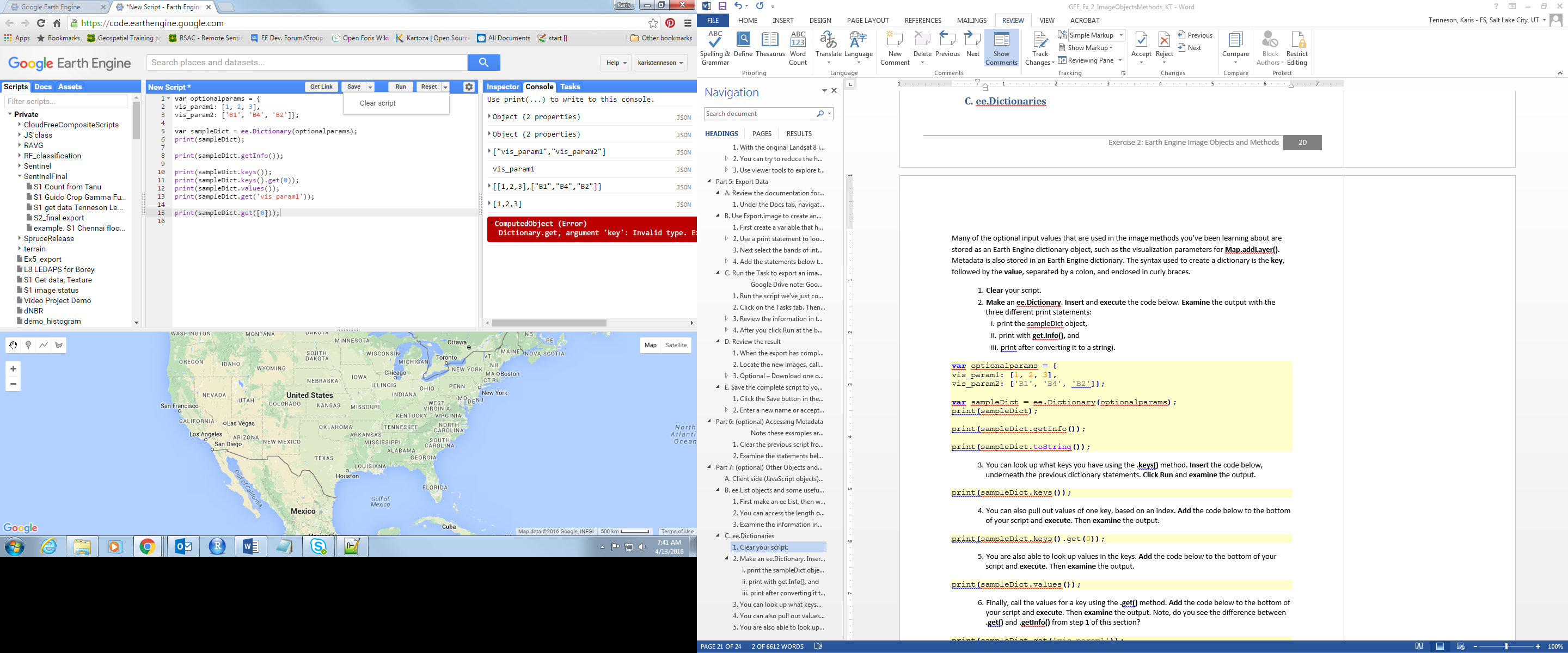
* 1. Những đối tượng Phía máy khách/Client side (đối tượng trong JavaScrips) và phía máy chủ/server side (đối tượng trong Earth Engine)

Những cấu trúc dữ liệu cơ bản khác của Earth Engine bao gồm **ee.Dictionary, ee.List, ee.Array, ee.Date, ee.Number** và **ee.String**. Rất nhiều trong số chúng có một bản sao tương tự trong JavaScript. Tuy nhiên, nếu bạn tạo một đối tượng JavaScript thay cho một đối tượng Earth Engine, bất kỳ một thao tác nào với đối tượng này đều sẽ được thực hiện tại máy khách/client side, trình duyệt web của bạn, sử dụng CPU máy tính của bạn. Tốt hơn là (trong phần lớn các trường hợp) sử dụng các đối tượng Earth Engine như vậy chúng sẽ được thực hiện trên máy chủ/server side (ví dụ, trên CPU của Google cloud). Các đối tượng Earth Engine (ví dụ **ee.List**) cũng tương tác một cách hiệu quả hơn với các đối tượng Earth Engine khác (ví dụ **ee.ImageCollection**). Trình duyệt của bạn, ví dụ, Chrome, không hiểu gì về các đối tượng Earth Engine trong script của bạn nếu bạn không yêu cầu thông tin cụ thể về chúng. Hàm **print()** hoặc **Map.addLayer()** yêu cầu khởi động (các mạch khởi động của máy tính) chuyển thông tin từ máy chủ của Google đến Code Editor rồi trình duyệt web của bạn.

* 1. Đối tượng ee.Number và một số hàm hữu ích

ee.Number là một đối tượng số học trong Earth Engine, đi kèm với một tập hợp các hàm liên quan, ví dụ hàm tính toán/arithmetic. Bạn sẽ tìm hiểu một phần của các hàm này, chúng rất hữu ích cho bạn trong tương lai khi làm việc với Earth Engine.

* + 1. **Xóa** script/tập lệnh của bạn.



* + 1. Tạo một đối tượng số học trong Earth Engine. **Copy** và **paste** những dòng lệnh dưới đây vào trình duyệt của bạn. **Chạy** những dòng lệnh này và **kiểm tra** kết quả.

**var** myNum **=** ee**.**Number**(**10.4**)** **;**

print**(**myNum**);**

* + 1. Bạn có thể làm tròn giá trị số học thành một số nguyên dương/whole number, một số nguyên/integer, bằng cách gõ **.round** ngay sau tên của biến chứa số này. **Thêm** dòng code dưới đây và **chạy** nó. **Kiểm tra** kết quả.

**var** round\_myNum **=** myNum**.**round**();**

print**(**round\_myNum**);**

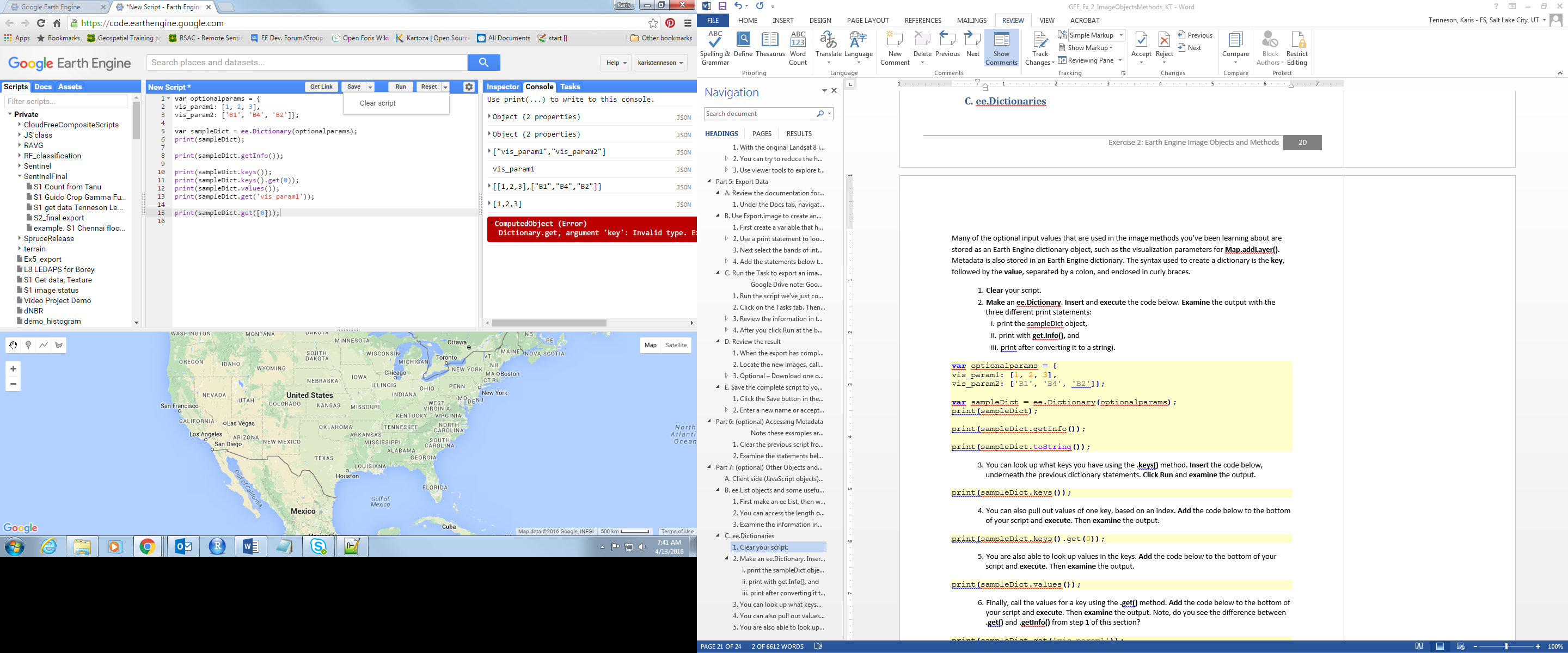
* + 1. **Tham khảo** bảng dưới đây để tìm hiểu một số hàm được dựng sẵn mà bạn có thể dùng cho đối tượng **ee.Number**. Tham khảo cột Cú Pháp Ví Dụ nếu cần thiết.Chú ý: Các biến, a và b, là một biến có giá trị số học trong Earth Engine.

|  |  |  |
| --- | --- | --- |
| Hàm | Chức Năng | Cú Pháp Mẫu |
| a**.**max(b**);** | Tìm giá trị lớn nhất của hai số, a và b. | **var** a **=** ee**.**Number**(**5.01**);**  **var** b **=** ee**.**Number**(**9**);**  print**(**a**.max**(b)**);**  //prints: 9 |
| a**.**min(b)**;** | Tìm giá trị nhỏ nhất của hai số, a và b. | **var** a **=** ee**.**Number**(**5.01**);**  **var** b **=** ee**.**Number**(**9**);**  print**(**a**.min**(b)**);**  //prints: 5.01 |
| a**.**int()**;** | Đưa giá trị số đầu vào về dạng số nguyên, a signed 32-bit integer. | **var** a **=** ee**.**Number**(**5.01**);**  print**(**a**.int**()**);**  //prints: 5 |
| a**.**sqrt()**;** | Tính căn bậc hai của một số (trong trường hợp này là a). | **var** a **=** ee**.**Number**(**9**);**  print**(**a**.sqrt**()**);**  //prints: 3 |
| a**.**log()**;** | Tính logarit tự nhiên của một số (trong trường hợp này là a). | **var** a **=** ee**.**Number**(**5.01**);**  print**(**a**.log**()**);**  //prints: 1.611… |

* 1. Đối tượng ee.String và một số hàm hữu ích

Chuỗi là một tập hợp các kí tự, và một ví dụ của string là từ “blue band”. Bạn sẽ tìm hiểu một phần các hàm thường được sử dụng và thuộc tính string trong Earth Engine.

* + 1. **Xóa** script của bạn.



* + 1. Tạo một đối tượng string Earth Engine. **Copy** và **paste** tập lệnh dưới đây vào trình duyệt của bạn. **Chạy/Execute** các dòng lệnh này và **kiểm tra** kết quả.

**var** StringVar **=** ee**.**String**(**"NDVI is a vegetation index"**);**print**(**StringVar**);**

* + 1. Bạn có thể tìm độ dài của string bằng cách gõ **.length()** ngay sau tên của biến string. **Đưa vào** và **chạy** dòng code dưới đây. **Kiểm tra** kết quả.

print**(**StringVar**.**length**());**

// returns how many characters the string contains

* + 1. **Tham khảo** bảng dưới đây để tìm hiểu một số hàm được dựng sẵn mà bạn có thể sử dụng với đối tượng **ee.String**. Tham khảo cột Cú Pháp ví dụ nếu cần thiết.Chú ý: biến, StringVar, là một biến chứa chuỗi ký tự/string trong Earth Engine.

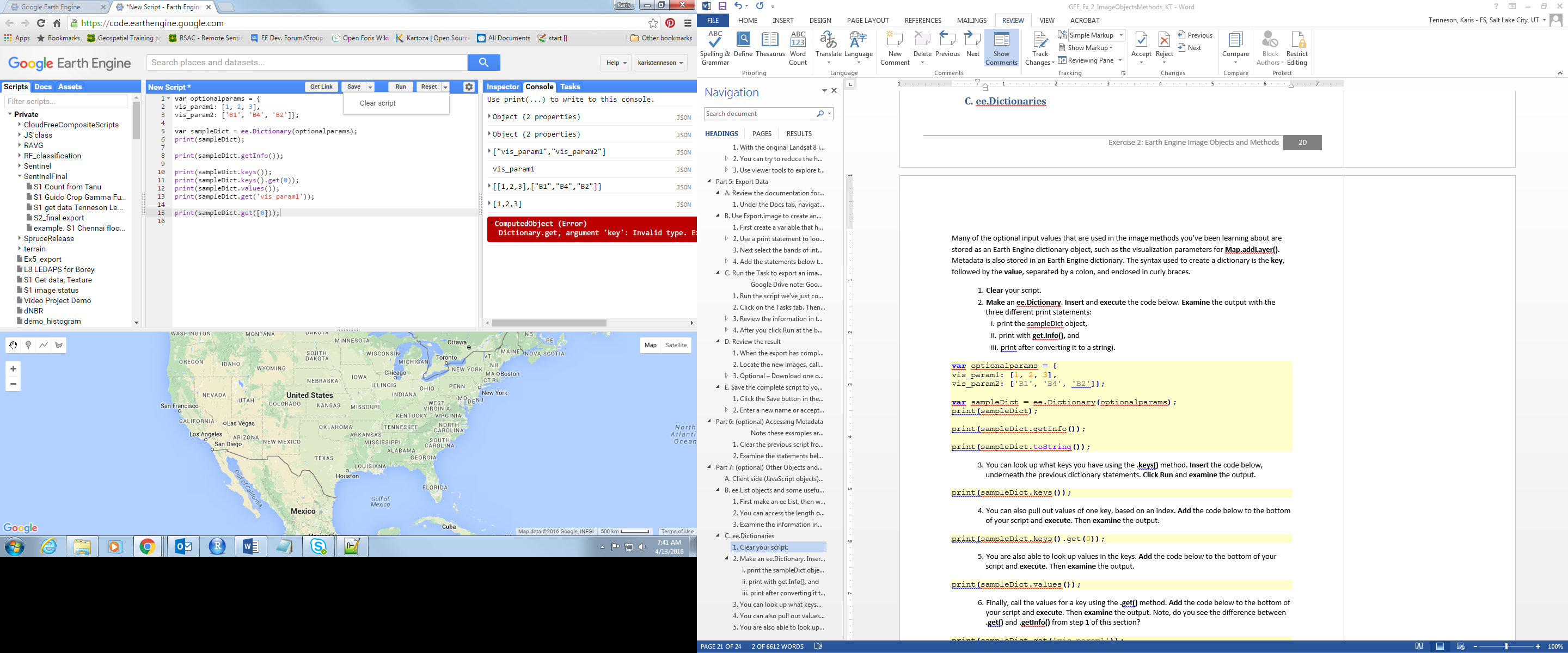
|  |  |  |
| --- | --- | --- |
| Phương Pháp | Chức năng | Cú Pháp Ví Dụ |
| StringVar**.**toUpperCase**();** | Chuyển chuỗi ký tự/string thành chữ in hoa. | **var** StringVar **=** ee**.**String('NDVI is a vegetation index')**;**  print**(**StringVar**.**toUpperCase**());**  // prints: NDVI IS A VEGETATION INDEX |
| StringVar**.**toLowerCase**();** | Chuyển chuỗi ký tự/string thành chữ in thường. | **var** StringVar **=** ee**.**String('NDVI is a vegetation index')**;**  print**(**StringVar**.**toLowerCase**());**  // prints: ndvi is a vegetation index |
| StringVar**.**split**();** | Phân tách chuỗi ký tự/string dựa vào dấu cách (hoặc với bất kỳ tham số đầu vào nào). | **var** StringVar **=** ee**.**String('NDVI is a vegetation index')**;**  print**(**StringVar**.**split**(**' '**));**  // prints: ["NDVI","is","a","vegetation","index"] |
| StringVar**.**index**();** | Tìm vị trí của ký tự đầu tiên trong chuỗi ký tự bạn quan tâm. Giá trị -1 sẽ xuất hiện nếu chuỗi ký tự bạn muốn tìm không nằm trong câu/cụm từ. Nhớ rằng, hàm sẽ đếm bắt đầu từ 0, không phải 1. Chú ý Index được dùng để phân biệt vị trí các chuỗi ký tự. | **var** StringVar **=** ee**.**String('NDVI is a vegetation index')**;**  print**(**StringVar**.**index**(**'vegetation'**));**  // prints: 10 |
| StringVar**.**slice**();** | Chia chuỗi ký tự tại vị trí nhất định. | **var** StringVar **=** ee**.**String('NDVI is a vegetation index')**;**  print**(**StringVar**.**slice**(**10**,**20**));**  // prints: vegetation |

* 1. Đối tượng ee.Date và một số hàm hữu ích

Cũng giống như chuỗi ký tự/strings and số học/numbers, JavaScript cũng có các đối tượng ngày/date, **Date();** tuy nhiên GEE lại có đối tượng date/ngày riêng, **ee.Date()**, nó được thiết kế cho môi trường làm việc của GEE. Tốt hơn là sử dụng đối tượng **ee.Date()** khi tương tác với ảnh và tập ảnh. GEE cũng có các đối tượng liên quan, **ee.DateRange()**, giúp làm việc với các tập hợp. Các đối tượng date/ngày của GEE thể hiện một thời điểm nhất định về thời gian và lưu trữ như một giá trị thời gian, độ chính xác tới một phần nghìn giây bắt đầu từ 1 tháng 1, 1970 UTC.

**Tại sao là 1 tháng 1, 1970?** Đó là ‘thời gian tham chiếu’. Trong JavaScript, đối tượng ngày/Date được lưu trữ chính xác đến một phần nghin giây từ ngày này. Đó là một cách chung để biểu diễn ngày trong giới lập trình.

* + 1. **Xóa** script của bạn.



* + 1. **Đưa vào/Insert** và **chạy/execute** dòng code dưới đây để thiết lập hai biến, startYear và startJulian (ngày của năm).

**var** startYear **=** 2015**;**

print **(**startYear**);**

**var** startJulian**=** 31**;**

print**(**startJulian**);**

* + 1. Bây giờ **thêm** dòng code dưới đây và **chạy/execute**. Dòng đầu tiên tạo ra ngày 1 Tháng 1, 2015, năm 2015 được khai báo cho biến startYear, tiếp theo, từ ngày đó tiến thêm 31 ngày bạn sẽ có startJulian day là 1 tháng 2, 2015. **Xem** kết quả từ câu lệnh print trong console. startDate được in dưới dạng yy-mm-dd hh:mm:ss (năm-tháng-ngày giờ-phút-giây). Nếu bạn mở rộng “Date” trong cửa sổ console bạn sẽ thấy giá trị của nó trong tham chiếu thời gian (1422748800000) mặc dù console hiển thị chúng dưới định dạng chúng ta có thể đọc được.

**var** startDate **=** ee**.**Date**.**fromYMD**(**ee**.**Number**(**startYear**),**1**,**1**)**

**.**advance**(**startJulian**,**'day'**);**

print**(**startDate**);**

* + 1. Bây giờ **thêm** và **chạy** code dưới đây. Dòng đầu tiên thiết lập một biến collection (tập hợp) mới. Dòng thứ 2 tạo một danh sách (list) từ khoảng ngày (date range) của tập hợp (collection’s date range). Dòng thứ 3 chuyển khoảng ngày (date range) thành một đối tượng ee.DateRange. **Xem** kết quả trong cửa sổ console sau khi lệnh print được thực hiện.

**var** collection **=** ee**.**ImageCollection**(**'COPERNICUS/S1\_GRD'**);**

**var** dates **=** ee**.**List**(**collection**.**get**(**'date\_range'**));**

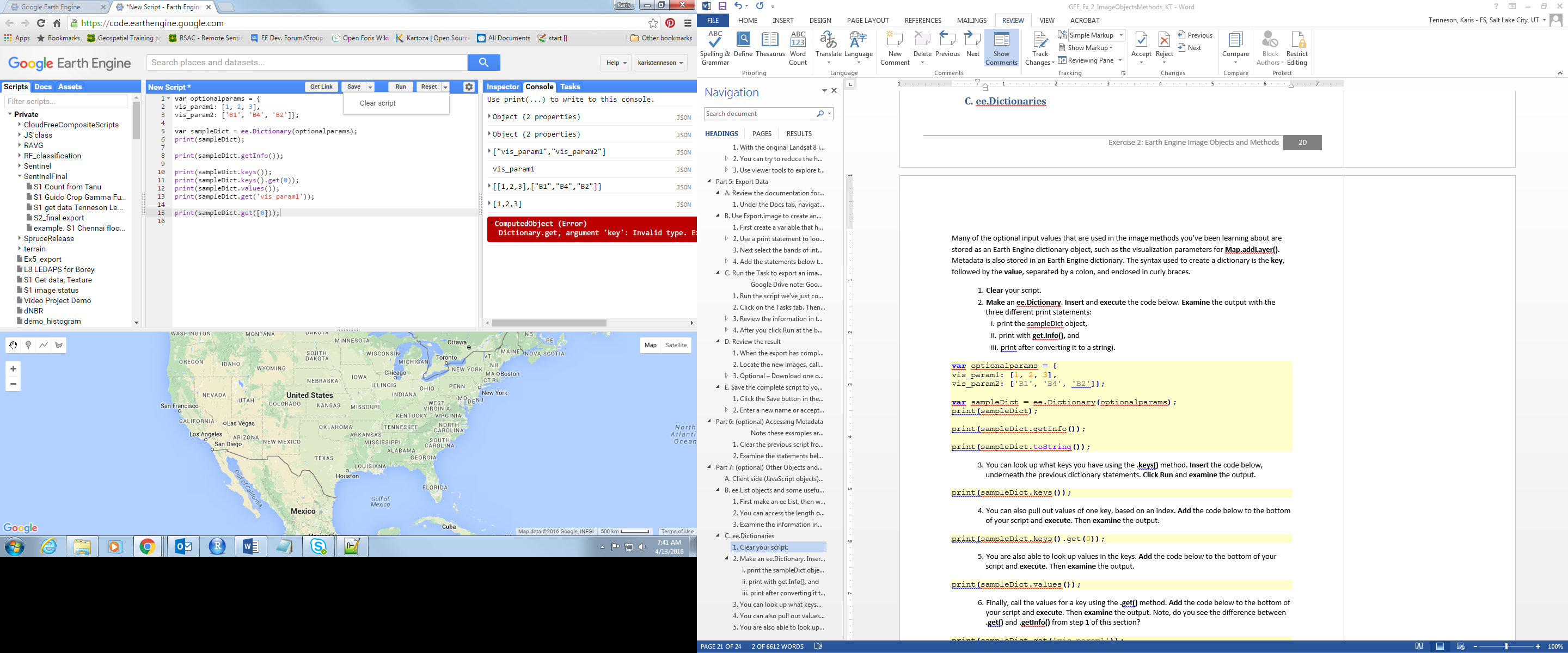
**var** dateRange **=** ee**.**DateRange**(**dates**.**get**(**0**),** dates**.**get**(**1**));**

print**(**'Date range: '**,** dateRange**);**

* 1. Các đối tượng ee.List và một số hàm hữu ích

Một danh sách theo thứ tự của các giá trị, chúng có thể là các đối tượng số học/numeric hoặc chuỗi/string. Đối tượng danh sách/list của Earth Engine có các thuộc tính và hàm riêng. Bạn sẽ tìm hiểu một phần của những hàm này, chúng sẽ rất hữu dụng khi bạn làm việc với ảnh trong Earth Engine.

* + 1. Trước tiên **Xóa** script của bạn.



* + 1. **Tạo** một **ee.List**, và bạn có thể tìm hiểu một số hàm ee.List. Thêm/**Insert** vày chạy/ **execute** dòng code dưới đây. **Kiểm tra** kết quả.

var myList = ee.List([10, 20, 50, 40, 130]);

print(myList);

* + 1. Bạn có thể tìm được độ dài của danh sách/list bằng cách gõ **.length()** ngay sau tên của list. **Thêm** và chạy/**execute** dòng code dưới đây. **Kiểm tra** kết quả đã được in.

print(myList.length());

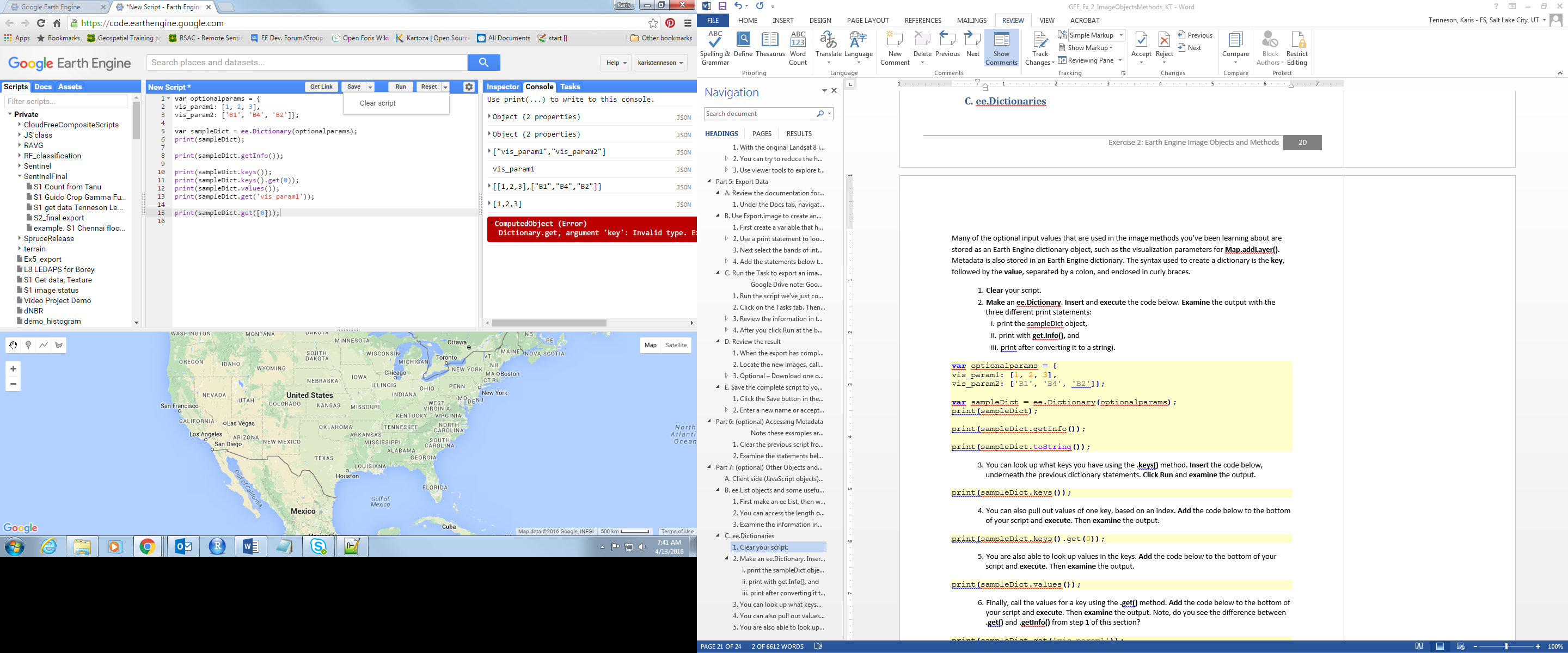
* + 1. **Tham khảo** bảng dưới đây để tìm hiểu một số hàm dựng sẵn mà bạn có thể sử dụng cho đối tượng **ee.List**. Tham khảo cột Cú Pháp Mẫu nếu cần thiết.Chú ý: biến, myList, đại diện cho các biến chứa danh sách/list trong Earth Engine.

|  |  |  |
| --- | --- | --- |
| Hàm | Chức năng | Cú Pháp Mẫu |
| myList**.**reverse**()** | Đảo thứ tự danh sách/list | **var** myList **=** ee**.**List**([**9**,** 12**,** 5**]);**  print**(**myList**.**reverse**());**  //prints: [5, 12, 9] |
| myList**.**length**()** | Đặt độ dài cho danh sách/list | **var** myList **=** ee**.**List**([**9**,** 12**,** 5**]);**  print**(**myList**.**length**());**  //prints: 3 |
| myList**.**sort**()** | Sắp xếp list/danh sách theo thứ tự | **var** myList **=** ee**.**List**([**9**,** 12**,** 5**]);**  print**(**myList**.**sort**());**  //prints: [5, 9, 12] |

* 1. ee.Dictionaries

Nhiều giá trị tham số tùy chọn sử dụng trong các hàm làm việc với ảnh mà bạn đã được học được lưu trữ tại đối tượng dictionary của Earth Engine, ví dụ như các thông số hiển thị cho **Map.addLayer()**. Metadata/Siêu dữ liệu cũng được lưu trong một Earth Engine dictionary. Cú pháp được sử dụng để tạo ra một dictionary là: một **khóa**, theo sau là **giá trị**, ngăn cách bởi dấu hai chấm, và đóng bằng dấu ngoặc cong }.

* + 1. **Xóa** script của bạn.



* + 1. **Tạo** một **ee.Dictionary**. **Thêm** và **chạy** dòng code dưới đây. **Tìm hiểu** đầu ra với ba lệnh in khác nhau:
       1. In đối tượng sampleDict,
       2. In với **get.Info(),** và
       3. In sau khi chuyển chúng thành một chuỗi/string).

**var** optionalparams **=** **{**

vis\_param1**:** **[**1**,** 2**,** 3**],**

vis\_param2**:** **[**'B1'**,** 'B4'**,** 'B2'**]};**

**var** sampleDict **=** ee**.**Dictionary**(**optionalparams**);**

print**(**sampleDict**);**

print**(**sampleDict**.**getInfo**());**

print**(**sampleDict**.**toString**());**

* + 1. Bạn có thể tra cứu những khóa/key bạn dã sử dụng hàm **.keys()**. **Thêm** code dưới đây, ngay phía dưới dòng lệnh dictionary. **Click Run** và **kiểm tra** kết quả.

print**(**sampleDict**.**keys**());**

* + 1. Bạn cũng có thể loại ra các giá trị của một khóa/key, dựa vào một chỉ số/index. **Thêm** dòng code dưới đây vào phần cuối cùng của script/tập lệnh của bạn và **chạy**. Sau đó **kiểm tra** kết quả.

print**(**sampleDict**.**keys**().**get**(**0**));**

* + 1. Bạn cũng có thể tra cứu các giá trị trong các khóa/key. **Thêm** dòng code dưới đây vào phần cuối cùng trong script của bạn và **chạy**. Sau đó **kiểm tra** kết quả.

print**(**sampleDict**.**values**());**

* + 1. Cuối cùng, gọi các giá trị cho một khóa/key sử dụng hàm **.get()**. Thêm/**Add** code dưới đây vào phía cuối script của bạn và **chạy**. Sau đó **kiểm tra** kết quả. Chú ý, bạn có thấy sự khác biệt giữa **.get()** và **.getInfo()** từ bước 1 của phần này?

print**(**sampleDict**.**get**(**'vis\_param1'**));**